Claims:

1. A compiling method including:

a first step of carrying out a syntax analysis of a description file describing a desired electronic circuit model with a predetermined high level description language, to generate a control data flow graph having a predetermined graph structure; and

a second step of dividing said control data flow graph into threads composed of a set of a plurality of connected nodes and achieving a particular function, and optimizing the divided threads to meet with a predetermined area restriction and a predetermined waiting time restriction, to obtain designation information of the number, the function, the placement and routing of logic cells for the desired electronic circuit model.

2. A compiling method claimed in Claim 1 wherein the optimization in said second step is carried out by estimating a minimum boundary of an area and a waiting time in connection with any of a function unit, a register and a multiplexor.

3. A compiling method claimed in Claim 1 wherein the optimization in said second step is carried out by first optimizing the divided threads to meet with the predetermined area restriction, and thereafter optimizing the optimized threads to meet with the predetermined waiting time restriction.

4. A compiling method claimed in Claim 3 wherein said second step includes:

a top-down processing step carrying out the optimization in connection with the predetermined area restriction and the predetermined waiting time restriction, in the order from a highest level divided thread; and

a down-top processing step of dividing a lower level divided thread optimized in said top-down processing step, into some number of threads, to assemble into a predetermined context or a predetermined circuit.

5. A compiling method claimed in Claim 4 wherein said top-down processing step includes:

a first dividing step for dividing the control data flow graph into threads composed of a set of the plurality of connected nodes and achieving the particular function;

a first scheduling step of allocating a predetermined control step and a thread moving range in that step for a thread obtained in the first dividing step, the first scheduling step also allocating the order of priority for the threads respectively allocated with the control steps, in accordance with a plurality of priority order lists previously set;

a first area restriction determining step for estimating a total area of the threads allocated in the first scheduling step, and of determining whether or not the estimated total area meets with the predetermined area restriction;

when it is determined in the first area restriction determining step that the estimated total area does not meet with the predetermined area restriction, a similarity cost calculating step for calculating a similarity cost in connection with an area for all thread pair combinations of the threads obtained in the first dividing step;

a first allocation step of selecting, from the thread pairs, a thread pair belonging to different control steps and having a further high similarity cost, with reference to the similarity costs obtained in the similarity cost calculating step, the first allocation step further obtaining a new thread by combining the selected thread pair as a new thread to another thread;

a second area restriction determining step for estimating a total area for the new thread pair obtained in the first allocation step, and of determining whether or not the estimated total area meets with the predetermined area restriction;

when it is determined in the second area restriction determining step that the estimated total area does not meet with the predetermined area restriction, an allocation-scheduling step of selecting, from the threads included in the list, a thread pair belonging to the same control step and having a further high similarity cost, in accordance with the plurality of priority order lists, in the order from a low priority list, the allocation-scheduling step obtaining a new thread pair by combining the selected thread pair as a new thread to another thread, and subdividing the control step allocated to the new thread pair, into two control steps having the same content;

when it is determined in the first or second area restriction determining step that the estimated total area meets with the predetermined area restriction, a thread processing step of investigating a trade-off between the area restriction and the waiting time restriction for the new thread pair obtained in the first allocation step or in the allocation-scheduling step, and carrying out the placement and routing of nodes to meet with both the restrictions, and

wherein said down-top processing step includes:

a second scheduling step of selecting and separating, for the threads placed and routed in the thread processing step, a thread pair having a low similarity, from the threads included in the list, in accordance with the plurality of priority order list, in the order from a high priority list; and

a second dividing step of assembling the thread pairs separated in the second scheduling step, into a context or a circuit which minimizes a connecting restriction between threads.

6. A compiling method claimed in Claim 5 wherein the predetermined time restriction in said thread processing step includes three restrictions, a movement range restriction defined as a movement range of the thread in said control step, a thread sharing restriction defined as an overlapping in time between the threads in said control step, and a pipeline restriction defined as a waiting time for the thread belonging to one loop of a pipeline processing executing said control step in parallel.

7. A compiling method claimed in Claim 5 wherein said thread processing step includes:

when new thread pairs obtained in said first allocation step or said allocation-scheduling step include a thread pair which does not meet with one of said movement range restriction, said thread sharing restriction and said pipeline restriction, a thread adjusting step for finding out a solution having a minimum thread area meeting with those restrictions for said thread pair; and

a thread optimizing step for investigating a critical path having a maximum delay, for a thread pair obtained in said thread adjusting step, on the basis of a predetermined connectivity restriction, to assemble nodes

into a cluster, when there exists a thread having a waiting time longer than a predetermined clock cycle, said thread optimizing step obtaining the number of registers to be inserted into said thread, and estimating a minimum area by timing said registers, thereby to obtain a solution meeting with said predetermined waiting time restriction.

8. A compiling method claimed in Claim 7 wherein said thread processing step includes:

a step of calculating a closeness matrix representing the closeness of nodes of each thread for the thread pair obtained in said thread adjusting step;

a step for generating a node cluster tree by grouping nodes based on said closeness matrix;

a step for investigating a critical path having a maximum delay on the basis of the connectivity metrics of each node pair in said node cluster tree; and

a step of grouping said node pairs included in said node cluster tree on the basis of whether or not the node pair belongs to said critical path, thereby to constitute an elementary block, and further grouping elementary blocks closest to each other, to constitute a macro block.

9. A compiling method claimed in Claim 7 wherein in said thread adjusting step and in said thread optimizing step, the step for finding out the solution is carried out by connecting a library supplying a set of function units which have corresponding area and delay and which have a predetermined parameter which can be set.

10. A compiling method claimed in Claim 5 wherein when the depth of at least one branch in a group of connected nodes exceeds a predetermined threshold value, the thread divided in said first dividing step is defined as a block which is found out between two continuous memory accesses or I/O accesses sharing the same I/O port, or as an express machine introduced by a user, or as a branch connecting node of said control data flow graph.

11. A compiling method claimed in Claim 10 wherein said control step includes a loop having a memory access, said thread found out between the continuous I/O accesses is given with a loop extension dependency for determining whether or not a memory parallel exists in the iteration of said loop.

12. A compiling method claimed in Claim 1 wherein a layout metrics for evaluating the area and the delay is used for the optimization in said second step.

13. A compiling method claimed in Claim 1 wherein said electronic circuit model is constituted of a hardware cell formed of a predetermined number of basic elements.

14. A compiling method claimed in Claim 13 wherein said hardware cell is one of an application specific integrated circuit, a field programmable gate array and a dynamic reconfigurable logic.

15. A synthesizing system including:

a front-end compiler means for carrying out a syntax analysis of a description file describing a desired electronic circuit model with a predetermined high level description language, to generate a control data flow graph having a predetermined graph structure; and

a back-end compiler means for dividing the control data flow graph into threads composed of a set of a plurality of connected nodes and achieving a particular function, and optimizing the divided threads to meet with a predetermined area restriction and a predetermined waiting time restriction, to obtain designation information of the number, the function, the placement and routing of logic cells for the desired electronic circuit model.

16. A synthesizing system claimed in Claim 15 wherein said back-end compiler means carries out optimization by estimating a minimum boundary of an area and a waiting time in connection with any of a function unit, a register and a multiplexor.

17. A synthesizing system claimed in Claim 15 wherein said back-end compiler means includes:

a first dividing means for dividing the control data flow graph into threads composed of a set of the plurality of connected nodes and achieving the particular function;

a first scheduling means of allocating a predetermined control step and a thread moving range in that step for a thread obtained in the first dividing means, the first scheduling means also allocating the order of priority for the threads respectively allocated with the control steps, in accordance with a plurality of priority order lists previously set;

a first area restriction determining means for estimating a total area of the threads allocated in the first scheduling means, and for determining whether or not the estimated total area meets with the predetermined area restriction;

when it is determined in the first area restriction determining means that the estimated total area does not meet with the predetermined area restriction, a similarity cost calculating means for calculating a similarity cost in connection with an area for all thread pair combinations of the threads obtained in the first dividing means;

a first allocation means of selecting, from the thread pairs, a thread pair belonging to different control steps and having a further high similarity cost, with reference to the similarity costs obtained in the similarity cost calculating means, the first allocation means further obtaining a new thread by combining the selected thread pair as a new thread to another thread;

a second area restriction determining means for estimating a total area for the new thread pair obtained in the first allocation means, and for determining whether or not the estimated total area meets with the predetermined area restriction;

when it is determined in the second area restriction determining means that the estimated total area does not meet with the predetermined area restriction, an allocation-scheduling means for selecting, from the threads included in the list, a thread pair belonging to the same control step and having a further high similarity cost, in accordance with the plurality of priority order lists, in the order from a low priority list, the allocation-scheduling means obtaining a new thread pair by combining the selected thread pair as a new thread to another thread, and subdividing the control

step allocated to the new thread pair, into two control steps having the same content;

when it is determined in the first or second area restriction determining means that the estimated total area meets with the predetermined area restriction, a thread processing means of investigating a trade-off between the area restriction and the waiting time restriction for the new thread pair obtained in the first allocation means or in the allocation-scheduling means, and carrying out the placement and routing of nodes to meet with both the restrictions;

a second scheduling means for selecting and separating, for the threads placed and routed in the thread processing means, a thread pair having a low similarity, from the threads included in the list, in accordance with the plurality of priority order list, in the order from a high priority list; and

a second dividing means for assembling the thread pairs separated in the second scheduling means, into a context or a circuit which minimizes a connecting restriction between threads.

18. A synthesizing system claimed in Claim 17 wherein the predetermined time restriction includes three restrictions, a movement range restriction defined as a movement range of the thread in said control step, a thread sharing restriction defined as an overlapping in time between the threads in said control step, and a pipeline restriction defined as a waiting time for the thread belonging to one loop of a pipeline processing executing said control step in parallel.

19. A synthesizing system claimed in Claim 17 wherein in said thread processing means carries out the placement and routing of said nodes, by connecting a library supplying a set of function units which have a predetermined area and a predetermined delay and which have a predetermined parameter which can be set.

20. A synthesizing system claimed in Claim 15 wherein said electronic circuit model is constituted of a hardware cell formed of a predetermined number of basic elements.

21. A synthesizing system claimed in Claim 20 wherein said hardware cell is one of an application specific integrated circuit, a field programmable gate array and a dynamic reconfigurable logic.

22. A recording medium recording a computer program for causing a computer to execute a processing for carrying out a syntax analysis of a description file describing a desired electronic circuit model with a predetermined high level description language, to generate a control data flow graph having a predetermined graph structure, and another processing for dividing the control data flow graph into threads composed of a set of a plurality of connected nodes and achieving a particular function, and optimizing the divided threads to meet with a predetermined area restriction and a predetermined waiting time restriction, to obtain designation information of the number, the function, the placement and routing of logic cells for the desired electronic circuit model.